CSE 332: Data Structures and Parallelism

Section 7: Parallel Primitives

0. Parallel Quicksort Recurrence

(a) Fill out the table below with each respective recurrence for the best case span and runtime.

Quicksort	Sequential Sort	Parallel Sort
Sequential Partition		
Parallel Partition		

(b) Which part of the recurrence is related to the Sequential Sort and explain why it is such?

(c) Which part of the recurrence is related to the Parallel Sort and explain why it is such?

(d) Which part of the recurrence is related to the Sequential Partition and explain why it is such?

(e) Which part of the recurrence is related to the Parallel Partition and explain why it is such?

1. River Con-current-ency

Batman and Robin share a jig of water. Rather than going to the store and buying a new bottle whenever they run out (so wasteful!), they continually refill the same jug, In addition, they keep track of how many cups are left in the jug on a sticky note so that they don't have to look in the jug before adding water to it or pouring themselves a delicious glass of water.

```
private int cupsOfWater = 0;
private int maximumWater = 8;
private Stack<Water> water;

private int checkWater() {
  return cupsOfWater;
}

public Water getWater() {
  if (checkWater() > 0) {
      return pourWater()
  }
  // cryDeeply()
}

public void addWater(Water e) {
  if (cupsOfWater != maximumWater) {
      water.push(e);
      cupsOfWater++;
  }
}
```

(a) Let's say that each person is a thread. Provide an interleaving of the two people that causes the maximum amount of embarrassment.

(b) Let's say we put a lock around every time cupsOfWater and water is called. Does this make our scenario thread-safe?

(c) Where would you put the lock to prevent concurrency problems?

(d) Due to budget cuts, we also share one singular cup now, and having learned our lesson from last time, we decide put a new lock every time someone tries to access this cup. What problems could this cause?